DOCUMENT RESUME

ED 209 229 SP 019 084

AUTHOR Welsh, William A.
TITLE An Adaptive Evaluation Structure for Computer-Based Instruction.
PUB DATE Oct 81
NOTE 27p.; Paper presented at the Annual Conference of the International Congress for Individualized Instruction (13th, Tucson, AZ, October 28-30, 1981).

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Branching; *Computer Assisted Instruction; *Computer Managed Instruction; *Computer Programs; Higher Education; Individualized Instruction; Student Evaluation
IDENTIFIERS *Adaptive Evaluation Structure

ABSTRACT

Adaptive Evaluation Structure (AES) is a set of linked computer programs designed to increase the effectiveness of interactive computer-assisted instruction at the college level. The package has four major features, the first of which is based on a prior cognitive inventory and on the accuracy and pace of student responses. AES adjusts materials presented to the student by using a branching process through a three-dimensional matrix of instructional blocks for each of four learning tasks. There are three levels of difficulty for each of the three dimensions--task complexity, degree of abstraction, and the extent of comparative analysis required. A second feature is that AES maps the path taken by each student through each task, records the time required as well as the number and matrix location of each appropriate and inappropriate response. Another feature of AES is that other student performance indicators can be stored for comparison with AES performance. A fourth feature is AES's ability to generate statistics on aggregate class use of and performance on each block. AES has been implemented with an instructional package called Comparative Study of Political Elites (COMPELITE), but is designed to be generalizable. It is written in BASIC V for PRIME minicomputers, and has been tested in one undergraduate political science course at the University of Iowa. (Author/FG)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED209229

October, 1981

AN ADAPTIVE EVALUATION STRUCTURE FOR COMPUTER-BASED INSTRUCTION

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

William A. Welsh

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

William A. Welsh
Professor of Political Science
Arizona State University*

*The Software described in this paper was developed at the Laboratory for Political Research, University of Iowa, under grant #SER 78-00701 from the Local Course Improvement program of the National Science Foundation. Programming and documentation were done by Douglas Dillard of the Laboratory for Political Research. The author was Principal Investigator.

SP 019 084

Prepared for presentation at the 13th Annual Conference of the International Congress for Individualized Instruction, Tucson, Arizona, October 28-30, 1981.

AN ADAPTIVE EVALUATION STRUCTURE FOR COMPUTER-BASED INSTRUCTION

WELSH, WILLIAM A.
Department of Political Science, Arizona State U., Tempe, AZ 85287

AES is a set of linked computer programs designed to increase the effectiveness of interactive CAI by (1) adapting the pace and content of instruction to the capabilities and performance of individual students, and (2) providing the instructor with a sensitive description of the student's CAI experience. The package has four main features. (1) Based on a prior cognitive inventory and on the accuracy and pace of student responses, AES adjusts the material presented to the student. It does this through a branching process based on decision rules which select a path through a 3-dimensional matrix of instructional blocks for each of four learning tasks. There are 3 levels of difficulty for each of the 3 dimensions -- task complexity, degree of abstraction, and extent of comparative analysis required. (2) AES maps the path taken by each student through each task, records time required as well as the number and matrix location of each appropriate and inappropriate response, and provides this information to the instructor on demand. (3) Other student performance indicators can be stored for comparison with CAI performance. (4) Statistics on aggregate class use of and performance on each block can be generated. AES has been implemented with one instructional package (COMPELITE: Comparative Study of Political Elites), but is designed to be generalizable. It is written in BASICV for PRIME minicomputers. It has been tested in one undergraduate political science course at Iowa.

paper

BACKGROUND AND PURPOSE

The software described in this paper was developed under a grant from the Local Course Improvement program of the National Science Foundation, and was designed to meet two related needs. Both needs have to do with instructional evaluation; one is perhaps more "substantive," the other, more "technical." First, there is a need to better evaluate the effectiveness of our efforts to impart basic analytical and critical thinking skills to students. Second, there is a need for more CAI packages which are dynamic and adaptive, tailoring the content of CAI experiences to the demonstrated capabilities and performance of individual students.

In the social sciences, the computer has served as an instrument of instructional evaluation in two limited senses. First, record-keeping software has been developed which records information such as the accuracy of student responses, the frequency with which students make use of CAI packages, and the differential responses of students to different types of questions. Used in this way, the computer is nothing more than a recording and tabulating device. Second, simple linear branching techniques have been used in CAI programming, based on the accuracy of student responses to specific questions. Interactive instructional packages routinely distinguish among correct and incorrect responses and provide the student with what amount to prompts to encourage more appropriate answers. In some programs, students are not permitted to move to another question or set of questions until they have offered acceptable responses to prerequisite items.

Interestingly, however, the initial pedagogic excitement with the "teaching machine" conception (or, "adaptive/reactive teaching system") (see Smallwood,

1962) has not been followed through on in the social sciences. Such a system would be adaptive, in the sense of interacting with the student in a way functionally analogous to that of a human instructor in a one-on-one learning situation, modifying responses in keeping with the student's evidenced capabilities as well as past and current performance. The system would also be self-evaluating and self-corrective, in the sense of assessing the effectiveness of its own components and making changes in the presentation of those components. For example, a self-corrective CAI package, on the basis of internal decision rules, might delete tutorial paths which were never used, or questions never answered correctly, or almost always answered correctly. At a minimum, such a system would provide such self-analysis to the course instructor.

I am not aware that such CAI packages exist in the social sciences. The software described in this paper does not fully meet the criteria advanced above, either. It does adapt its responses on the basis of student capabilities and performance. And it does provide internal program diagnostics to the course instructor. It does not yet make internal changes automatically.

Most tutorial and data analysis CAI packages in the social sciences are largely insensitive to variations among student users, in two senses. First, most programs move students through tutorial content or exercises in a linear and undifferentiated way. There is one basic sequence of material, and all students pass through that sequence sooner or later, provided they finish the task at all. Second, in the few existing packages which try to take student performance into account, all students begin with material of the same level of difficulty, and the branching in the program is based on a simple bifurcation between correct and incorrect, or acceptable and unacceptable, responses.

The likelihood that different students begin with substantially varying capabilities, and thus logically should be started with material of differential content and level of difficulty, is not attended. Further, existing CAI packages do not take into account different patterns of cognition--different ways of thinking, analyzing, and inferring that in fact characterize the student population with which we work.

It is also the case that our past efforts have given inadequate attention to the importance of designing CAI software useful in evaluating the contribution of CAI to the imparting of analytical and critical thinking skills on the part of students. Presumably, one of the most important goals of education in the social sciences is the development of precisely such skills. More specifically, we try to teach skills in problem definition, the framing of hypotheses, the examination and simple analysis of data, the systematic inference of results from data, and the ability to develop or refine generalizations based on a set of experiences confronting data. Unfortunately, we do not have effective ways of knowing whether, or to what degree, these skills are being imparted. Relatedly, we do not have sensitivity to the differences among individual students in their abilities to develop such skills. Finally, we have little systematic evidence, and not much more intuitive grasp, about the relationships between the logical and cognitive capabilities that students bring to us, on the one hand, and the impact of computer-assisted instruction, on the other hand.

We know that the cognitive and logical capabilities of our students vary considerably. For example, students vary in their ability to deal with abstractions; some grasp generalizations quickly, whereas others require varying degrees of concretization. Of those who depend on concretization,

some learn "programmatically" (through repetition and sequential handling of structured tasks), whereas for others the critical element of understanding is what has been called "cultural code"--the linking of abstractions to examples of circumstances with which they are familiar in everyday life. Another dimension of cognitive variation is the extent to which a student can learn individually, as opposed to through associative, interactive processes. Other dimensions of cognitive difference include differences in the ability to identify patterns of similarity and difference in sets of objects, to see relationships expressed in various forms, and to deduce conclusions by linking premises and evidence.[3]

The usual approaches to identifying cognitive differences--where such efforts are in fact made--have been based on conventional inventory instruments, e.g., questionnaires. A basic characteristic of such instruments is that they attempt to assess cognitive complexity and logical skills through a clearly contrived inventory structure. They are set stimuli situations which permit student responses only within a specific, and often narrowly-conceived, set of parameters defined by the questions which are being asked. There is no a priori reason to reject such contrived instruments, but it seems likely that we will learn a good deal more about students' cognitive and logical capabilities as we unstructure our inventorying instruments and allow the student to reveal the "pathways" of thinking most congenial to him or her. One approach to doing this is computer mapping of the logical "path" taken by students through a relatively open, multi-dimensional instructional "space." Such a strategy underlies the software described below.

A final word of background is necessary about the different types of CAI

materials now in use in the social sciences. These materials may be roughly divided into three categories: tutorials, data analysis packages, and simulations. It is not our purpose here to define formally or to illustrate comprehensively each type; we need only distinguish among the principal pedagogic contributions of each. Tutorials, consisting of text, questions, and responses, are designed for the presentation of new concepts and/or drill on a reinforcement of ideas previously presented to the student. Data analysis packages generally are intended to teach problem definition and problem formulation, inference from evidence, and synthetic abilities. Data analysis software may also have the goal of providing some hands-on experience with specific techniques of data manipulation and analysis, e.g., descriptive or theoretical statistics. Simulations teach projective/analytical skills, largely by cultivating the student's ability intelligently to ask, "What would happen if. . .?"

One of the most important characteristics of CAI development in the social sciences is that these three basic modes of CAI are almost always used singly, rarely in combination. The development of the software described in this paper proceeded in part from the assumption that the pedagogic contributions of any instructional mode might well be enhanced by its use in direct conjunction with one or both of the other modes. As a tentative first step, then, the software described here combines the tutorial and data analysis modes. It attempts to provide a hands-on opportunity for students to apply and test, through simple analysis of a large data set, the ideas presented in the tutorial portions of the package. The tutorial and data analysis modes are fully integrated; that is, the program moves back and forth between tutorial, question-and-answer sequences, and data analysis and inference tasks.

STRUCTURE OF AES

The software is called the Adaptive Evaluation Structure (AES). Although AES is a software structure--i.e., a linked set of programs which are content-independent--it is difficult to describe AES without also describing COMPELITE (Comparative Analysis of Political Elites), the substantive learning package which is the first flesh put onto the bone structure that is AES.

COMPELITE was developed originally at the Laboratory for Political Research at the University of Iowa (where development of AES also took place) as an interactive learning package for the Hewlett-Packard 2000 and 3000 series minicomputers. The data set on which COMPELITE is based includes 17 items of information on more than 2,000 political elites from seven countries. The data were originally gathered as part of four quite separate studies, conducted at different times and for somewhat different purposes. As a result, considerable effort was involved in devising functionally equivalent measures valid across the several country studies. (The method used is described in Przeworski and Teune, 1970.) The data generally include social background, career, attitudinal, and behavioral measures.

The original version of COMPELITE was far simpler than its AES-based successor. The initial version was segmented into several "variable sets," i.e., subsets of the 17 variables comprising the full data set. There were 19 (obviously overlapping) subsets. Each variable set included measures that would permit the student to structure tables on the basis of which a simple descriptive or analytic question could be answered. The variable sets did not have to be used in sequence, although the Student User's Guide recommended a sequence running from simple comparative descriptions ("What differences can

be observed among the seven countries in the social origins of their elites?") through simple bivariate relationships ("What significance do education and training have for the subsequent characteristics of an elite's career?") to three-way cross-classifications with one or more control variables ("How does social background affect the relationship between career characteristics and the likelihood that an elite will exhibit an ideological posture in policy preferences? Do these relationships vary between countries?"). There was no interactive tutorial material, and the program itself did not probe the student's understanding of the tables he or she had constructed. That was left for more conventional classroom settings. In short, the original version of COMPELITE was a nonadaptive, nonevaluational data analysis package. Our aim in integrating it into AES was to make it adaptive, tutorial, and evaluational.

AES is structured around a set of n => 2 tasks. We have used four tasks in COMPELITE, but are not aware of limitations on the number of separate analysis tasks, except for core or on-line storage limitations at any given installation. AES is written in BASIC V for PRIME minicomputers. I have asked Arizona State University for resources to have the package converted to the PDP 11/70 during the second semester of 1981-82. Some modifications are being made to the PRIME version at the University of Iowa, and I would expect that that version would be available for distribution by mid-winter, 1982.

Each task consists of a 3 x 3 x 3 matrix of instructional "blocks." This means that, as now structured, each task may have a maximum of 27 distinct blocks of material for presentation to the student. (There are no program-imposed limitations on the size of each block.) In fact, we have implemented COMPELITE with 10 blocks per task, as described below. Blocks may be added if our experiences with student users suggest the need for more complex branchings.

Each block contains material directed at most or all of the following learning functions: tutorial presentation and drill, problem definition, problem formulation, table creation, data analysis, and data interpretation/inference.

The 3 x 3 x 3 matrix structure implies that each block of instructional material is assigned a level of difficulty rating (1 = easy; 2 = average; 3 = challenging) on each of the three dimensions. These dimensions correspond to the three types of cognitive and logical skills we are trying to impart to students: task complexity, degree of abstraction, and extent of comparison required. Broadly, the abstraction dimension embodies skills in problem definition and problem formulation. The task complexity dimension refers to task operationalization (i.e., table construction) and the ability to see basic relationships when represented in tabular form. The comparative dimension reflects skills of recognizing patterns of similarity and difference, and drawing meaningful inferences from such evidence.

Some illustration might help to clarify the nature of these three dimensions, and of the distinctions among levels of difficulty for each dimension. This illustration is provided in Figure 1.

(Insert Figure 1 About Here.)

The resulting nature of the matrix of instructional blocks for each task is depicted in Figure 2. I discuss criteria for entry into, and branching rules for movement through, each task below. At this point, we should simply note that the general purpose is to move the student from Block 1,1,1 (or higher, in the case of students whose prior Cognitive Inventory scores suggest that they should be able to start at a more advanced point on one or more dimensions) to Block 3,3,3--or, as close to that point as their capabilities and the program-imposed time limits for each task permit. As the program is

Level of Difficulty	(1) Task Complexity	(2) Abstraction	(3) Comparison
1	Simple 2-way tables	Variables given, expected direction of relationship specified	1 country, given
2	Compare (2) 2-way tables	Variables selected by student from several (given) possibilities, direction of expected relationship indirectly implied	2 countries, must be appropriately selected by student (criteria are general)
3	3-way tables (control variable used)	Abstract statement of purpose of analysis only given	3 countries to be selected by student; criteria are restrictive, but only implied by the nature of the analytic task

Figure 1: Bases for Distinguishing among Levels of Difficulty for Three Learning Dimensions

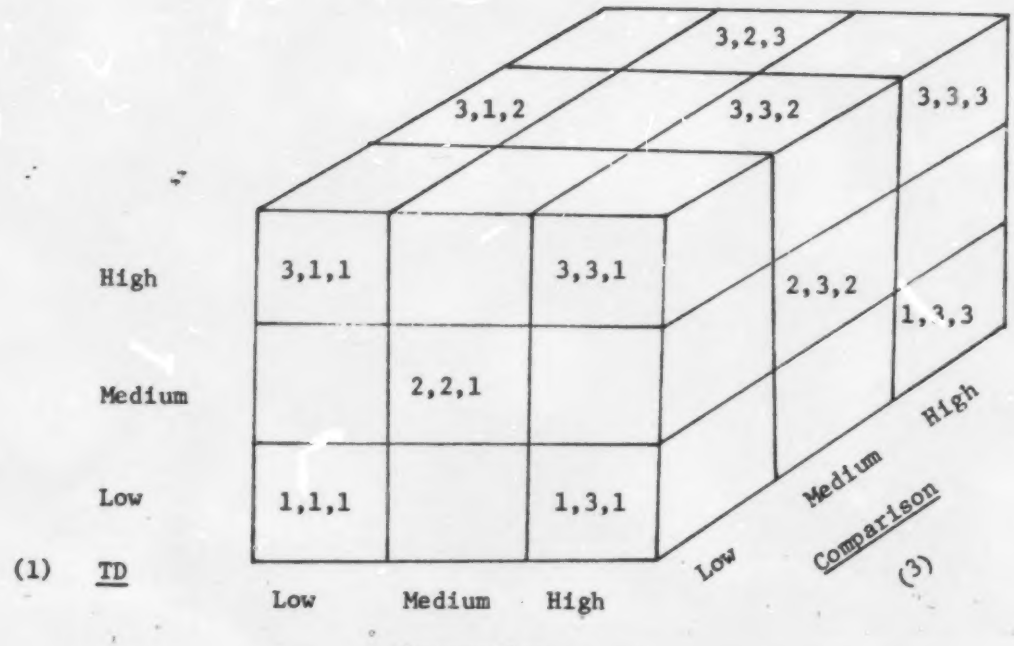

Figure 2: 3-Dimensional Task Matrix
(numbered blocks for illustration only)

now written, a student may not be advanced or moved backward more than one level of difficulty at a time. One of the refinements planned for incorporation into AES during the next few months is an accelerator factor for demonstrably superior students.

(Insert Figure 2 About Here.)

The block locations, represented by their conjoint position on the three dimensions just described, plus a task identification number assigned by the program, are the bases for the AES mapping function, described in the next section of the paper. This mapping function is critical to our ability to understand better how students are able to refine and apply their cognitive and logical capabilities.

Progression And Branching Rules. Understanding the mechanics of AES requires answers to some additional questions. How is it decided where students enter AES/COMPELITE in the first place? What are the constraints on forward and backward movement along each dimension which collectively define parameters for the range and nature of the "paths" that students may take through the task matrix? How is a student's work on a given task terminated? Similarly, when is an instructional session ended by AES?

The initial entry point for a student is determined by his or her score on a Cognitive Inventory instrument administered at the beginning of the instructional term. The CI score is stored by AES both as a single overall score, and as separate scores for each of the three cognitive/logical dimensions. The Cognitive Inventory instrument was developed as part of the NSF/LOCI project grant, on the basis of a thorough search through existing cognitive inventory instruments. The difficult--and still problematic--part of building the CI was deciding which inventory items should be used to index

which of the three dimensions. (The CI instrument is available to other institutions. I would be pleased to have formal testing or informal evaluations done of it at other locations.) The CI scores for any given group of students are then grouped into three categories, corresponding to the three levels of difficulty for each of the cognitive/logical dimensions. At present we simply divide the student population into thirds on each dimension, but a simple program change would allow the instructor to use more sophisticated criteria for this categorization. We do not feel that we have enough solid evidence about our students, and about the CI instrument itself, as yet to justify inserting a more complex procedure. We do plan to incorporate a minimum-threshhold criterion into our version of AES shortly.

The CI dimension scores are then used to determine the entry point for each student's first use of AES/COMPELITE. Note that the three dimensions are treated discretely; a student whose CI scores indicate he or she should do well on task complexity, average on abstraction, and less well on comparative analysis would be started at Block 3,2,1, for example.

The CI scores for each dimension are the basis for calculating the constantly updated Current Aptitude (CA) scores. The CA scores are major factors in determining where the student will enter the next task. In calculating the CA scores, the exact path taken by each student through each task is recorded. The difficulty level for each dimension of each block presented to the student is compared with the difficulty levels for the immediate preceding block. If the student moved to a more difficult block, a "+" is assigned. If he or she moved backward to a less difficult block on that dimension, a "-" is assigned. If the block to which the student moved was at the same level of difficulty for a given dimension, a "0" is

assigned. These signs are coupled with the difficulty level itself; that is weighted difficulty scores are created. This means that success or failure on a more difficult block affects the CA score more than does success or failure on a less difficult block. The weighted difficulty scores are then summed, at the end of each task, and added to the raw CI scores, or to the current CA scores, to produce updated CA scores for each dimension.

We also calculate Entry Level Success scores (ELSS), for each dimension and each task. These scores indicate the difficulty level of the first block used successfully by the student at his or her most recent session with AES/COMPELITE. "Success" is defined as moving to a more difficult block without having to go back to an easier block. Thus if a student entered at level 3 of a given dimension, and did not have to go back to a level 2 block on that dimension, his or her ELSS for that dimension would be "3." We do not permit the ELSS to fall below 1 level less than the initial entry point. Thus if the student starts at level 3, his or her ELSS could not be set lower than "2," even if the student actually was required to go back to a level 1 block on that dimension.

The CA scores and the ELSS are used to determine where a student will enter a subsequent block. The two measures are used as follows.

1. If ELSS = CA level, student enters at this level.
2. If ELSS - CA level = 1 or -1, then the student enters at whichever score (ELSS or CA) is lower.
3. If ELSS - CA level = 2 or -2, the student enters at level 2 for that dimension.

In short, when the difference between ELSS and the CA level is small (only one level), the student is moved down to the lower level. If the

magnitude of the difference between ELSS is larger (two levels), this obviously indicates a substantial discrepancy between the "reading" offered by the two scores, and recommends that the student be started at the middle level of difficulty on that dimension on the next task.

In advancing a student within a task, the level of difficulty is first incremented for task complexity (D1), then for abstraction (D2), and finally for comparison (D3). This procedure is based on the premise that problem formulation and table set-up are the least difficult skills to master, whereas making meaningful inferences across (substantially different) cases is the most difficult. This is my intuitive sense from teaching these skills to undergraduate students for 17 years, but it is one that is very much open to challenge. One of the very useful things about the tracking/mapping function of AES is that it provides an evidential base for evaluating this premise. Apparent "intransitivities," or failures to scale, among blocks with prior ordering are easy to identify from the AES diagnostics.

This incrementing procedure means that the "pure" case of forward movement through any given task--that is, assuming the student entered at 1,1,1 and handled acceptably each block presented--would be:

1,1,1 --> 2,1,1 --> 2,2,1 --> 2,2,2 --> 3,2,2 --> 3,3,2 --> 3,3,3.

The pure progressive set thus requires only seven of the potential 27 block locations in the task matrix.

In decrementing blocks (that is, in moving students backwards), the same substantive principle is applied. That is, the first dimension on which the student should "retreat" from a higher level of difficulty is the dimension of comparative analysis. Decrementing proceeds on a given dimension until the student has, if necessary, fallen back to level one on that dimension

before decrementing on the other dimensions is done. For example:

2,2,2 --> 2,2,1 --> 2,1,1 --> 1,1,1

also

2,3,3 --> 2,3,2 --> 2,3,1 --> 2,2,1 --> etc.

In general, failure to successfully complete a block results in backward movement. However, two points of qualification should be made. First, some blocks contain within themselves prompts to encourage correction of initially unsatisfactory student responses. A single "error" does not necessarily mean that the block will be failed. Second, for many of the blocks there is a range of acceptable responses. Some may be more acceptable than others, and this is conveyed to the student in the program's response. But a philosophy of "let the student see the implications of his or her response" prevails in some parts of AES/COMPELITE.

Incidentally, an inspection of all possible decremented block locations which could be needed--assuming that a student could fall all the way back to 1,1,1 from any of the block locations that could be achieved during forward movement--reveals that three additional block locations must logically be present. These are 3,1,1; 3,2,1; and 3,3,1. Empirically, it has turned out that the 10 resulting block locations have been adequate for all purposes in the limited tests so far conducted with AES/COMPELITE. However, it is possible that CI, CA and/or ELSS scores could call for a student to enter a block location not now used. The program handles this by moving the student to the closest higher block.

There are three ways in which a student may complete work on a given task.

1. By completing Block 3,3,3 successfully. This moves the student directly on to the next task.

2. By completing at least Block 2,2,2 successfully, but using 1.96 standard deviation units more than the class average time for that task. (A default value is entered until at least 10 uses of the task have been recorded by class members.) Here the result is being moved on to the next block.

3. By completing at least Block 2,2,2 successfully, but using 1.96 standard deviation units more than the class average time on two tasks within the same terminal session. In this case, the session is terminated, but the student will progress on to the next task at his or her next terminal session. (The choice of this rather permissive time criterion can be changed in the program.)

In addition to the contingency described just above in (3), there are two other ways in which a session may be terminated.

1. If a student fails twice in the same session to complete Block 2,2,2 of a given task.

2. If a student fails to progress as far as Block 2,2,2 of a given task within 1.96 standard deviation units more than the class average time for that task.

Question Types and Functions. There are four types of questions presently used in AES. Each is directed toward at least slightly different instructional goals, and the four exhibit some variation in the forms of responses that are called for and accepted. Type 1 is a relatively simple multiple-choice question. In this question type, several possible answers are available, and more than one answer can be treated as correct by the program. Further, regardless of the adequacy of the student response, the student may be branched to any of the available next questions (forward or backward, of course). Thus,

the instructor has full flexibility in determining how to treat a given student response, and to where that response should next lead the student. In this respect, AES Type 1 questions are somewhat more flexible than multiple-choice questions found in many canned "coursewriter-type" programs.

A Type 2 question is designed to produce a crosstabulation setup and run. The student must supply the names of variables and cases (in COMPELITE, countries) to be used for the crosstab. Selection (always a country) and control variables are available (but optional from the program's point of view.) The instructor supplies a list of valid variable and case selection responses against which the student input is matched. If permitted to by the instructor, the program will automatically prompt for categories of cases and/or variables--e.g., a student asked to provide the name of a pluralist country who has not done so in setting up his or her crosstab can be prompted with the word, "pluralist." The program also has validity checks for more complicated case selection situations, such as when the instructor wishes to allow the student to choose a case from any of n categories of cases, but wishes to limit the number of cases selected to < n. Generally, Type 2 questions are used in assessing the student's ability to translate a substantive problem into operational form.

Type 3 questions are a special form of multiple-choice question linked to an immediately preceding Type 2, i.e., crosstabulation. These questions involve probes of the student's understanding of the crosstabulation he or she has just constructed. Because the choice of variables and countries for tables in the more difficult blocks is relatively wide, the possible resulting combinations of cases and variables can be very large. The critical problem posed for programming was that the correct, or acceptable, answers to a given

question necessarily varied, depending on the countries and variables selected. It soon became clear that the original programming strategy of asking that the instructor provide discrete valid response lists for all combinations was unworkable. The program now has optional Boolean operators (.AND., .OR., .NOT.) to define combinations and subsets of cases and variables. This change made this section of the program itself a good deal longer and more complicated, but dramatically simplified text preparation for the instructor.

Because of the complexity of the response evaluation for some Type 3 questions, these questions must recognize four types of student responses. "True" answers are both valid (acceptable cases and variables were chosen) and interpretively correct (the inferences drawn from the table are defensible). "Null" answers are valid, but always interpretively incorrect; that is, the interpretation represented by this answer is inaccurate for any combination of variables and countries. "Illegal" answers refer to responses not presented by the program as alternatives. "False" answers are valid, and would be interpretively correct for some combination of cases and variables, but are not interpretively correct for the cases and variables chosen by the student in constructing his or her crosstabulation.

It is apparent that, in order for the program to handle the complex response evaluations often necessary for Type 3 questions, it must store and look up the cases and variables chosen for the preceding crosstabulation(s).

Type 4 questions accept "open-ended" responses, although an effort is made to guide the general direction of student entries. The program procedure is a general dump of student input into a file, which the instructor then can list out and evaluate. Generally, these questions are of a high enough degree of complexity that either (a) no branching decisions are necessary--e.g., at

the last question in Block 3,3,3 of a task; or (b) the specification of branching rules genuinely sensitive to the substance of the response would consume uneconomic amounts of program space. Most of these questions in AES/COMPELITE deal with possible interaction effects of a third variable on the relationship between two other variables. Because such questions are used at advanced performance levels, where student choice of cases and variables is deliberately broad, and because the discussion of interaction effects requires detailed treatment of specific categories within variables (variable values), the number of possible evidential combinations is so large as to defy reasonable a priori specification. Pedagogically, we also find that students are more willing to attempt to answer these complex queries when they understand that the instructor will personally look at their efforts--presumably with a degree of flexibility and imagination that the computer might not have invoked.

PROGRAM FILES AND THE EVALUATION FUNCTION

The AES package contains four programs and eight files. The specific functions of each program, and the contents of each file, are not of consequence for this paper. (They are described in Author's Manual and Program Description, NSF/LOCI grant SER 78-00701, available from the author.) Our concern is with the Student History file and the Task History file, since they provide the bases for the evaluation of the educational effects of AES/COMPELITE.

The Student History file contains a record for each time a student completes a task. (Modifications are now being made such that partial, unsuccessful uses of a task also will be recorded.) The record contains

the student's identification number, Cognitive Inventory (CI) and Current Aptitude (CA) scores for each cognitive/logical dimension, the date of the terminal session, the total time required for work on the task, and a complete mapping of the student's "path" through the blocks presented. From this information we can learn several important things. Illustrative of the kinds of questions that can be asked are:

1. With what kinds of questions does the student have difficulty? Which of the three dimensions of cognitive/logical capacity is most troublesome for the student?

2. What relationship exists between the student's scores on the initial Cognitive Inventory instrument, on the one hand, and the student's ability to handle increasingly difficult problems on each of the three dimensions? Do the CI scores "predict" student movement through the task matrix? If not, what implications might this finding have for the presumed validity of the CI scores?

3. Is there observable progress (by inference, "learning") in the student's ability to handle more difficult challenges along each cognitive/logical dimension as he or she completes more CAI tasks?

4. Relatedly, are there minimum and maximum "threshhold" levels of performance for a given student on each of the three dimensions? Do these threshholds change with increasing exposure to CAI?

5. Are the "paths" taken by a student similar across tasks? If not, we may be able, through careful examination of the substance of each task, to generate hypotheses about the effects of varying substantive content and examples on the performance of students' cognitive and logical skills.

6. Is there a relationship between the time required to complete a task, and the "path" taken through that task? Do shorter task times necessarily

involve fewer "failed" blocks? Or are those whose pace is slower, and sometimes more reflective, more likely to achieve more regular progression through the material? (Such a finding might prompt us to impose minimum response times on some questions, for example.) Is "trial-and-error" a necessary learning device for some students, whereas others view failure, even on isolated questions, as largely discouraging? Is there a complex, interactive relationship between CI/CA scores and task times as they jointly affect the success of a student's CAI performance?

7. When used in conjunction with other (non-CAI) measures of class performances--which it will be possible to store in the next version of AES--the contents of the Student History file can help illuminate the relationships between CAI performance and other measures of student achievement. Reflection on such relationships might well lead to refinement of more traditional classroom learning strategies, as well as of the CAI software.

8. The contents of the Student History file also reveal possible "intransitivities," or improper difficulty level ratings, among the instructional blocks. To cite a simple example, we might discover that many students consistently pass without difficulty blocks assigned level 3 difficulty in abstraction, but consistently fail certain blocks assigned level 2 difficulty on that same dimension. A more complicated empirical "scaling" of the blocks in a given task could be undertaken, using the contents of the Student History file and the Task History file.

The Task History file contains aggregated information about the class use of each block and each task. The basic information consists of:

1. The Block Frequency Count--How many times is each block in each task used?

2. The Block Failure Count--How many times has each block been failed? From items (1) and (2), the program can calculate the Block Failure Ratio, which is (2)/(1)*100--the proportion of time the block has been failed.

3. The Task Usage Count--How many times has each task been completed?

4. The Total Task Time Count--For how many total minutes has this task been used? From items (3) and (4), the program can calculate the average time required to complete a given task. This number is the basis for criteria used in terminating a student's work on a particular task and in terminating a session.

The contents of the Task History file, in addition to contributing to the identification of possible errors in the assignment of difficulty levels to given blocks, can be used for more traditional item analyses of specific blocks. The Task History file will be the basis of decisions concerning needed revisions, additions, and/or deletions of block and task structure.

AES allows us to do some important things that previous CAI packages in the social sciences did not permit. In particular, AES is distinguished by (1) the potential for substantially greater adaptability in adjusting the pace and content of instruction to the past and current performance of the individual student, and (2) the potential for a sensitive evaluation of the relationships among students' cognitive and logical capabilities, CAI, and more traditional modes of learning. A good deal of refinement remains to be done on AES/COMPELITE, and it obviously will be important to develop other substantive components to place into the AES structure. But the package already represents an important step forward. Some of the needed refinements must await more systematic testing of the package in classroom settings at Arizona State, at Iowa, and then at other universities.

That the evaluation we have done on AES so far is inadequate is especially clear when one reflects on the desiderata of CAI impacts. Broadly, we presumably should be able to expect at least five positive effects of the introduction of adaptive, evaluation-oriented CAI packages in given instructional settings:

1. Student information/comprehension levels should increase.

2. Students' cognitive and logical capabilities should be enhanced; they should be more confident masters of the learning process.

3. The new software should enhance the effectiveness of non-CAI materials used in the course.

4. The new software should improve the instructor's performance in the course.

5. The new software should provide a systematic basis for evaluation of itself by the course instructor. Ideally, the software should identify and initiate refinements in part automatically.

In short, new computer-based instructional technology cannot be judged solely on the basis of its technical sophistication, or the information it can provide, or the ease and clarity of its use by students--although these considerations obviously are important. The ultimate justification and growth of this technology depends on positive evaluations of its overall impact on specific instructional settings. We are, unfortunately, not very far down the road toward making those sorts of evaluations.

REFERENCES

Przeworski, A. and H. Teune. 1970. The Logic of Comparative Social Inquiry. New York: John Wiley.

Smallwood, R. D. 1962. A Decision Structure for Teaching Machines. Cambridge: M. I. T. Press.